

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A method of rendering a web page, the method comprising the computer-implemented steps of:

generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, wherein the mapping is stored separate from web pages associated with the one or more page parameters;

wherein said web pages include the web page;

wherein the web page has a designed hard-coded structure that includes a reference to a portlet that generates a component of the web page;

after the web page has been designed and stored, receiving a request to display the web page;

in response to receiving ~~[[a]]~~ the request to display the web page, performing the steps of:

determining that the web page is associated with a page parameter from the one or more page parameters;

retrieving and inspecting the mapping to determine that the page parameter is mapped to a portlet parameter of ~~[[a]]~~ the portlet that generates ~~[[a]]~~ the component of the page that is based, at least in part, on the portlet parameter;

wherein the portlet is executable code that is operable to generate page components;

passing a value associated with the page parameter as a value of the portlet parameter to the portlet that generates the component of the web page;

the portlet generating the component based upon the value associated with the  
portlet parameter; and

inserting the component that was generated by the portlet into the web page;  
wherein the step of inserting the component includes inserting the component  
at the location, in the structure of the web page, of the reference to the  
portlet;

wherein the steps of the method are performed by one or more computing devices.

2. (Currently Amended) The method of Claim 1, further comprising the step of mapping the page parameter, wherein mapping the page parameter comprises the steps of: mapping the page parameter to a second portlet parameter associated with a second component of the web page; and passing the value associated with the page parameter as the value of the second portlet parameter to a second portlet that generates the second component.
3. (Currently Amended) The method of Claim 1, further comprising the steps of: establishing a plurality of page parameters for the web page; and mapping the plurality of page parameters to a plurality of portlet parameters associated with the component of the web page; wherein the step of inspecting the mapping further comprises the step of inspecting the mapping to determine which page parameters of the plurality of page parameters are mapped to each of the plurality of portlet parameters; wherein the step of passing the value further comprises the step of passing, based on the mapping, values associated with the plurality of page parameters as the

values of the plurality of portlet parameters to the portlet that generates the component; and

wherein the step of the portlet generating the component further comprises the step of the portlet generating the component based upon the values associated with the plurality of portlet parameters.

4. (Currently Amended) The method of Claim 1, further comprising the step of mapping the page parameter to the portlet parameter associated with the component of the web page without mapping the page parameter to portlet parameters associated with any other components of the web page.
5. (Currently Amended) The method of Claim 1, further comprising the steps of mapping the page parameter to the portlet parameter and mapping a second page parameter to a second portlet parameter of the portlet that generates the component of the web page.
6. (Previously Presented) The method of Claim 1, further comprising the step of establishing for the page parameter a default value, and wherein the step of passing the value associated with the page parameter further comprises the step of passing the default value as the value of the portlet parameter to the portlet that generates the component.
7. (Currently Amended) The method of Claim 1, wherein the request to display the web page includes a URL and the URL includes the value associated with the page

- parameter, and wherein the step of passing the value associated with the page parameter is performed by passing the value contained in the URL as the value of the portlet parameter.
8. (Currently Amended) The method of Claim 1, further comprising the steps of:  
presenting to a user a user interface for customizing the web page;  
in response to the user interacting with the user interface, obtaining a user specified value for the page parameter; and  
wherein the step of passing the value associated with the page parameter is performed by passing the user specified value as the value of the portlet parameter to the portlet that generates the component.
9. (Previously Presented) The method of Claim 1, wherein a plurality of values are specified for the page parameter and wherein:  
the method further comprises the step of determining a selected value from the plurality of values based on an override hierarchy; and  
the step of passing further comprises the step of passing the selected value as the value of the portlet parameter to the portlet that generates the component.
10. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a URL page parameter value and a customized page parameter value and the override hierarchy specifies that the URL page parameter value is the selected value.

11. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a default page parameter value and a customized page parameter value and the override hierarchy specifies that the customized page parameter value is the selected value.
12. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a default page parameter value and a portlet specified value and the override hierarchy specifies that the default page parameter value is the selected value.
13. (Original) The method of Claim 1, further comprising the step of presenting to a page designer a user interface for specifying the mapping between the page parameter and the portlet parameter.
14. (Previously Presented) The method of Claim 1, further comprising the step of registering the portlet with a portal repository, wherein the process of registering the portlet causes data associated with the portlet to be stored in the portal repository.
15. (Previously Presented) The method of Claim 14, wherein the data associated with the portlet is communicated to the portal repository as an XML document.
16. (Previously Presented) The method of Claim 1, further comprising the step of receiving input from a page designer, through a user interface, to create the mapping between the portlet parameter and the page parameter.

17. (Previously Presented) The method of Claim 1, wherein the value associated with the page parameter is stored in memory and wherein:  
the method further comprises the step of retrieving the stored value; and  
the step of the portlet generating the component further comprises the step of the portlet generating the component based upon the retrieved value.
18. (Currently Amended) A method comprising the computer-implemented steps of:  
generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, wherein the first mapping is stored separate from web pages associated with the one or more page parameters;  
wherein the web pages include a web page;  
wherein the web page has a designed hard-coded structure that includes a reference to a portlet that generates a component of the web page;  
wherein, when the web page is rendered, the component generated by the portlet is inserted in the web page at the location, in the structure of the web page, of the reference to the portlet;  
in response to a user manipulating ~~[[a]]~~ the component ~~associated with a~~ of the web page, ~~[[a]]~~ the portlet that previously generated the component generating a particular event;  
wherein the portlet is executable code that is operable to generate page components;  
the portlet passing data that represents the particular event to logic associated with the web page;

retrieving and inspecting the first mapping that maps events to actions and event  
output parameters to page parameters;  
determining, based on the first mapping and the passed data, an action to perform in  
response to the particular event;  
inspecting the first mapping to determine that an event output parameter associated  
with the particular event is mapped to a page parameter; and  
causing the action to be performed, wherein causing the action to be performed  
comprises passing a value of the event output parameter as the value of the  
page parameter;  
wherein the steps of the method are performed by one or more computing devices.

19. (Currently Amended) The method of Claim 18, wherein:  
the web page is a first page and the page parameter is associated with a second page;  
and  
the step of causing the action to be performed further comprises the step of passing  
the value of the page parameter to logic responsible for rendering the second  
page.
20. (Previously Presented) The method of Claim 18, wherein the step of causing  
the action to be performed further comprises the step of generating a request that  
specifies a URL, wherein the value of the page parameter is included in the URL.
21. (Original) The method of Claim 20, wherein:

the step of generating the request further comprises the step of generating a request  
for executable code; and

the step of causing the action to be performed further comprises the step of invoking  
the executable code.

22. (Original) The method of Claim 21, wherein the executable code is a web  
service.

23. (Previously Presented) The method of Claim 18, wherein:  
the action comprises rendering a second page, wherein the page parameter is  
associated with the second page, and wherein rendering the second page  
comprises the steps of:  
inspecting a second mapping to determine that the page parameter is mapped  
to a portlet parameter of a second portlet that generates a second  
component of the second page that is based, at least in part, on the  
portlet parameter;  
passing the value of the page parameter as the value of the portlet parameter to  
the second portlet;  
the second portlet generating the second component based upon the value  
associated with the portlet parameter; and  
inserting the second component that was generated by the second portlet into  
the second page.

24-46. (Canceled)



47. (Previously Presented) The method of Claim 1, wherein the portlet is a first portlet and wherein the mapping maps a single page parameter, of the one or more page parameters, to a first portlet parameter of the first portlet and to a second portlet parameter of a second portlet.
48. (Canceled)
49. (New) A computer-readable volatile or non-volatile medium storing one or more sequences of instructions for rendering a web page, which instructions when executed by one or more processors cause performance of steps comprising:  
generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, wherein the mapping is stored separate from web pages associated with the one or more page parameters;  
wherein said web pages include the web page;  
wherein the web page has a designed hard-coded structure that includes a reference to a portlet that generates a component of the web page;  
after the web page has been designed and stored, receiving a request to display the web page;  
in response to receiving the request to display the web page, performing the steps of:  
determining that the web page is associated with a page parameter from the one or more page parameters;  
retrieving and inspecting the mapping to determine that the page parameter is mapped to a portlet parameter of the portlet that generates the component based, at least in part, on the portlet parameter;

wherein the portlet is executable code that is operable to generate page components;  
passing a value associated with the page parameter as a value of the portlet parameter to the portlet that generates the component of the web page;  
the portlet generating the component based upon the value associated with the portlet parameter; and  
inserting the component that was generated by the portlet into the web page;  
wherein the step of inserting the component includes inserting the component at the location, in the structure of the web page, of the reference to the portlet.

50. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of mapping the page parameter, wherein mapping the page parameter comprises:  
mapping the page parameter to a second portlet parameter associated with a second component of the web page; and  
passing the value associated with the page parameter as the value of the second portlet parameter to a second portlet that generates the second component.
51. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of:  
establishing a plurality of page parameters for the web page; and

mapping the plurality of page parameters to a plurality of portlet parameters

associated with the component of the web page;

wherein the step of inspecting the mapping further comprises the step of inspecting

the mapping to determine which page parameters of the plurality of page parameters are mapped to each of the plurality of portlet parameters;

wherein the step of passing the value further comprises the step of passing, based on

the mapping, values associated with the plurality of page parameters as the values of the plurality of portlet parameters to the portlet that generates the component; and

wherein the step of the portlet generating the component further comprises the step of

the portlet generating the component based upon the values associated with the plurality of portlet parameters.

52. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of mapping the page parameter to the portlet parameter associated with the component of the web page without mapping the page parameter to portlet parameters associated with any other components of the web page.
53. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of mapping the page

parameter to the portlet parameter and mapping a second page parameter to a second portlet parameter of the portlet that generates the component of the web page.

54. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of establishing for the page parameter a default value, and wherein the instructions that cause performance of the step of passing the value associated with the page parameter further comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the default value as the value of the portlet parameter to the portlet that generates the component.
55. (New) The computer-readable medium of Claim 49, wherein the request to display the web page includes a URL and the URL includes the value associated with the page parameter, and wherein the step of passing the value associated with the page parameter is performed by passing the value contained in the URL as the value of the portlet parameter.
56. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of:  
presenting to a user a user interface for customizing the web page;  
in response to the user interacting with the user interface, obtaining a user specified value for the page parameter; and

wherein the step of passing the value associated with the page parameter is performed  
by passing the user specified value as the value of the portlet parameter to the  
portlet that generates the component.

57. (New) The computer-readable medium of Claim 49, wherein a plurality of values are specified for the page parameter and wherein:

the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of determining a selected value from the plurality of values based on an override hierarchy; and

the instructions that cause performance of the step of passing the value associated with the page parameter further comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the selected value as the value of the portlet parameter to the portlet that generates the component.

58. (New) The computer-readable medium of Claim 57, wherein the plurality of values includes a URL page parameter value and a customized page parameter value and the override hierarchy specifies that the URL page parameter value is the selected value.

59. (New) The computer-readable medium of Claim 57, wherein the plurality of values includes a default page parameter value and a customized page parameter value and the override hierarchy specifies that the customized page parameter value is the selected value.

60. (New) The computer-readable medium of Claim 57, wherein the plurality of values includes a default page parameter value and a portlet specified value and the override hierarchy specifies that the default page parameter value is the selected value.
61. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of presenting to a page designer a user interface for specifying the mapping between the page parameter and the portlet parameter.
62. (New) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of registering the portlet with a portal repository, wherein the process of registering the portlet causes data associated with the portlet to be stored in the portal repository.
63. (New) The computer-readable medium of Claim 62, wherein the data associated with the portlet is communicated to the portal repository as an XML document.
64. (New) The computer-readable medium of Claim 49, further comprising the step of receiving input from a page designer, through a user interface, to create the mapping between the portlet parameter and the page parameter.

65. (New) The computer-readable medium of Claim 49, wherein the value associated with the page parameter is stored in memory and wherein:  
the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of retrieving the stored value; and  
the instructions that cause performance of the step of the portlet generating the component further comprise instructions which, when executed by the one or more processors, cause the portlet to generate the component based upon the retrieved value.
66. (New) A computer-readable volatile or non-volatile medium storing one or more sequences of instructions which, when executed by one or more processors, cause performance of steps comprising:  
generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, wherein the first mapping is stored separate from web pages associated with the one or more page parameters;  
wherein the web pages include a web page;  
wherein the web page has a designed hard-coded structure that includes a reference to a portlet that generates a component of the web page;  
wherein, when the web page is rendered, the component generated by the portlet is inserted in the web page at the location, in the structure of the web page, of the reference to the portlet;

in response to a user manipulating the component of the web page, the portlet that  
previously generated the component generating a particular event;  
wherein the portlet is executable code that is operable to generate page components;  
the portlet passing data that represents the particular event to logic associated with the  
web page;  
retrieving and inspecting the first mapping that maps events to actions and event  
output parameters to page parameters;  
determining, based on the first mapping and the passed data, an action to perform in  
response to the particular event;  
inspecting the first mapping to determine that an event output parameter associated  
with the particular event is mapped to a page parameter; and  
causing the action to be performed, wherein causing the action to be performed  
comprises passing a value of the event output parameter as the value of the  
page parameter.

67. (New) The computer-readable medium of Claim 66, wherein:  
the web page is a first page and the page parameter is associated with a second page;  
and  
the instructions that cause performance of the step of causing the action to be  
performed further comprise instructions which, when executed by the one or  
more processors, cause performance of the step of passing the value of the  
page parameter to logic responsible for rendering the second page.
68. (New) The computer-readable medium of Claim 66, wherein the instructions that



cause performance of the step of causing the action to be performed further comprise instructions which, when executed by the one or more processors, cause performance of the step of generating a request that specifies a URL, wherein the value of the page parameter is included in the URL.

69. (New) The computer-readable medium of Claim 68, wherein:

the instructions that cause performance of the step of generating the request further comprise instructions which, when executed by the one or more processors, cause performance of the step of generating a request for executable code; and the instructions that cause performance of the step of causing the action to be performed further comprise instructions which, when executed by the one or more processors, cause performance of the step of invoking the executable code.

70. (New) The computer-readable medium of Claim 69, wherein the executable code is a web service.

71. (New) The computer-readable medium of Claim 66, wherein:

the action comprises rendering a second page, wherein the page parameter is associated with the second page, and wherein rendering the second page comprises:  
inspecting a second mapping to determine that the page parameter is mapped to a portlet parameter of a second portlet that generates a second

component of the second page that is based, at least in part, on the  
portlet parameter;  
passing the value of the page parameter as the value of the portlet parameter to  
the second portlet;  
the second portlet generating the second component based upon the value  
associated with the portlet parameter; and  
inserting the second component that was generated by the second portlet into  
the second page.

72. (New) The computer-readable medium of Claim 49, wherein the portlet is a first portlet and wherein the mapping maps a single page parameter, of the one or more page parameters, to a first portlet parameter of the first portlet and to a second portlet parameter of a second portlet.